

# 证 明

本证明之附件是向本局提交的下列专利申请副本

申 请 日： 2002 12 05

申 请 号： 02 1 54006.3

申 请 类 别： 发明

发明创造名称： 视频点播系统的信道合并方法

申 请 人： 国际商业机器公司

发明人或设计人： 裴云彰； 邵凌； 谢东

中华人民共和国  
国家知识产权局局长

王 景 川

2003 年 5 月 12 日

## 权 利 要 求 书

1. 一种用于视频点播系统的信道合并方法, 其特征在于所述方法包括步骤:

5 (1) 响应于多个客户端对某一视频节目的请求, 建立一根信道 (S1) 和至少一子信道 (S11), 所述根信道 (S1) 根据最早发出请求的客户端的请求而建立, 所述多个子信道 (S11) 中的每一个对应于一较晚发出请求的客户端的请求而建立;

10 (2) 对于所建立的每一个信道, 监测使用该信道的客户端的数目变化, 如果使用被监测信道的客户端数目不为零, 则维持该信道, 如果使用被监测信道的客户端数目为零, 则关闭该信道。

2. 如权利要求 1 所述的信道合并方法, 其特征在于: 所述根信道 (S1) 和每个子信道 (S11) 的建立是响应于客户端的播放开始请求或节目跳转请求。

15 3. 如权利要求 1 所述的信道合并方法, 其特征在于所述根信道与多个子信道形成树结构。

4. 如权利要求 1 所述的信道合并方法, 其特征在于所述步骤 (2) 包括:

(2-1) 用一计数参数标记使用每个信道的客户端的数量;

20 (2-2) 响应于所述每个信道及其子信道的合并、跳转、或停止事件的发生, 减小所述计数参数的值;

(2-3) 如果所述计数参数的值等于零, 则在服务器端关闭所述信道。

5. 如权利要求 4 所述的信道合并方法, 其特征在于: 如果所述计数参数的值不等于零, 则在服务器端保持所述信道, 而执行了合并、跳转、或停止事件的所述客户端则不再接收所述信道播放的节目。

25 6. 如权利要求 1 所述的信道合并方法, 其特征在于步骤 (1) 包括步骤:

(1-1) 从所述子信道 (S11) 可能并入的根信道 (S1) 的集合中搜索一根信道 (S1), 所述根信道 (S1) 满足条件:  $\min(S11.start\_time - S1.start\_time) < object\_length/2$ , 其中,  $\min(S11.start\_time - S1.start\_time)$  表示所述子信道 (S11) 的开始时刻 (S11.start\\_time) 与待选根信道集合中的各个根信道的开始时刻 (S1.start\\_time) 的差值中的最小值, 30  $object\_length/2$  表示所播放的视频节目的总长度的 1/2;

(1-2) 如果所述根信道(S1)存在,则在所述根信道(S1)的子孙信道的集合中为所述子信道(S11)搜索一个其将要并入的父信道(S6),所述父信道满足条件:  $\min(S11.start\_time-S6.start\_time) < S6.start\_time-S5.start\_time$ , 其中,  $S6.start\_time$  表示所述父信道(S6)的开始时刻,  $S5.start\_time$  表示所述父信道(S6)的父信道(S5)的开始时刻;

7. 如权利要求6所述的信道合并方法,其特征在于:如果在步骤(1-1)中未找到所述根信道(S1),则将所述子信道(S11)设置为新的根信道,并将它的根信道参数设置为1,并且所述子信道(S11)是所述客户端收看的唯一信道。

8. 如权利要求6所述的信道合并方法,其特征在于:如果在步骤(1-2)中找到了所述父信道,则所述客户端同时收看所述子信道(S11)及所述父信道的视频节目。

9. 如权利要求6所述的信道合并方法,其特征在于:如果在步骤(1-2)中未找到所述父信道,则将所搜索到的根信道做为所述子信道(S11)的父信道,并且所述客户端同时收看所述子信道(S11)及所述根信道的视频节目。

10. 如权利要求2所述的信道合并方法,其特征在于:如果所述客户端在时刻  $t$  的请求是开始请求,则在子信道(S11)中将一开始时间参数设置为  $t$ 、将一对象偏移参数设置为零。

11. 如权利要求2所述的信道合并方法,其特征在于:如果所述客户端在时刻  $t$  的请求是跳转请求,且所述跳转的对象偏移时间为  $s$ ,则在所述子信道(S11)中将一开始时间参数设置为  $t$ 、将一对象偏移参数设置为  $s$ ,同时对所述客户端在之前时间接收的从时刻  $t$  开始播放的所述视频节目的信道执行停止操作。

12. 如权利要求5所述的信道合并方法,其特征在于:如果所述停止操作是所述视频节目已经结束,则直接关闭所述子信道并释放所述子信道的全部资源。

13. 如权利要求6所述的信道合并方法,其特征在于:在建立所述子信道(S11)的次级子信道时,重复步骤(1-1)和(1-2)以将所述子信道(S11)

确定为该次级子信道的父信道,并将所述子信道(S11)的计数参数的值加1。

14. 一种用于视频点播系统的信道合并装置,所述信道合并装置配置于

所述视频点播系统中的视频服务器中或与其可操作地连接，其特征在于所述信道合并装置包括：

- 信道选择单元，用于响应多个客户端对某一视频节目的请求，建立一根信道（S1）和至少一子信道（S11），所述根信道（S1）根据最早发出请求的客户端的请求而建立，所述多个子信道（S11）中的每一个对应于一较晚发出请求的客户端的请求而建立；

信道控制单元，用于对所建立的每一个信道，监测使用该信道的客户端的数目变化，如果使用被监测信道的客户端数目不为零，则维持该信道，如果使用被监测信道的客户端数目为零，则关闭该信道。

- 10 15. 如权利要求 14 所述的信道合并装置，其特征在于所述信道控制单元还包括：

计数单元，用一计数参数标记使用每个信道的客户端的数量；

- 15 其中当所述每个信道及其子信道发生合并、跳转、或停止事件时，所述计数单元减小所述计数参数的值；并且如果所述计数参数的值等于零，则在服务器端关闭所述信道。

16. 如权利要求 15 所述的信道合并装置，其特征在于：如果所述计数参数的值不等于零，则在服务器端保持所述信道，而执行了合并、跳转、或停止事件的所述客户端则不再接收所述信道播放的节目。

## 说明书

### 视频点播系统的信道合并方法

#### 5 技术领域

本发明一般地涉及信道合并方法,更具体地涉及在通信网络中利用传送视频流的多个信道的合并向多个客户端进行优化的组播(multicast)传送。

#### 背景技术

- 10 随着互联网使用的爆炸性增长和计算机能力的不断增加,人们对所谓的视频点播(video-on-demand)的应用的兴趣也极大地增长了,其中客户可以在任意时间请求媒体文件(视频、音频、数据等),用于即时观看或将来观看。但是,视频点播也提出了新的挑战,即服务器带宽和网络带宽的巨大消耗。一般地,每个请求都由一个专用的单播(unicast)流来支持,并且
- 15 基于单播的视频点播系统的开销非常庞大。信道合并技术的出现为视频点播服务创造了一个全新的模式,其目标就是通过使客户端同时接收两个或更多的视频流来减少满足请求特定视频对象的各客户端所需要的服务器带宽。当客户端为了立即观看的目的而接收数据并且同时进行存储时,服务器可以通过组播的方式使一个视频对象服务于多个用户,并因此既减少了网络带宽,
- 20 又减少了服务器带宽。

- 现有的信道合并方法可以分为三类:静态广播方法、合并树结构方法、和事件驱动(event driven)方法。由摩天楼(Skyscraper)代表的静态广播方法用指定的时段和长度、在几个信道中广播多段被请求的对象。静态广播的优点是在非常繁忙的环境中具有其简易性和相对高的效率。但是,当系
- 25 统的负载不高,或由于其不灵活的资源分配而使不同视频节目的普及度较分散时,静态广播的性能很差。对于由Dyadic代表的合并树结构方法,当有新的用户加入时,该合并树结构用表示信道的树节点动态地构建合并树,在用户真正需要信道之前不实际分配信道。这种方法通过消除闲置信道资源的浪费而克服了静态广播的缺陷。但是,当合并树仅由新用户的加入时间来确
- 30 定时,它就不直接支持诸如VCR(盒带录象机)功能,即任意停止、暂停、快进/倒退等。对于由SRMT(Simple Reachable Merge Target,简单可到达

合并目标) 和 CT (Closest Target, 最近目标) 代表的事件驱动方法, 当客户向服务器表示要进行播放、停止、跳转或合并操作时, 该方法动态地确定一组该客户应该使用的信道。因为每个客户的合并路径根据用户的交互作用而得到动态地调整, 所以该方法可以支持 VCR 功能。

5 下面将结合图 1、图 2 和图 3 对两个信道进行合并的方法说明如下:

第一步, 视频服务器 1 从客户端 A 接收播放视频的请求, 并根据该请求, 用信道 S6 向客户端 4 发送被请求的视频节目。

第二步, 当在一段时间 (T) 后从客户端 B 接收与所述客户端 A 相同的视频点播请求时, 该视频服务器 1 建立信道 S11, 并通知客户端 B 准备在信道 S11 和信道 S6 两个信道上从该视频服务器 1 接收该视频节目。

第三步, 该视频服务器 1 利用信道 S11 从该视频节目的开始点 (a) 向客户端 B 发送该视频节目, 并且客户端 B 进行接收, 同时客户端 B 在信道 S6 上与客户端 A 同步接收并存储从该视频服务器 1 继续发送的随后所述视频节目。

15 第四步, 视频服务器 1 将信道 S6 作为信道 S11 的父信道 (即信道 S11 将要并入的信道), 当客户端 B 在信道 S11 上接收的视频节目达到其在信道 S6 上接收并存储的视频节目的起始点 (b) 时, 即又经过了时间 T 时, 将信道 S11 并入信道 S6, 并且该视频服务器 1 将关闭信道 S11, 并停止在信道 S11 上向客户端 B 发送该视频节目。此时, 客户端 B 中已经存储了从点 (b) 到点 (c) 的视频节目。该信道 S11 (子视频流) 与其父信道 S6 合并后, 如果没有其它的客户端正在使用子信道 S11 (即正在使用信道 S11 的子信道), 那么这个子信道 S11 将被停止。

第五步, 在将信道 S11 并入信道 S6 后, 客户端 B 在用所述信道 S6 继续从点 (c) 接收并存储从视频服务器 1 发送的随后所述视频节目的同时, 以先进先出的方式从点 (b) 读取并播放已经存储在其本地存储器中的该视频节目, 使得该视频节目在客户端 B 上的播放能够连续进行。

25 尽管用于信道合并的事件驱动方法是控制组播信道的最灵活的方法, 但是这种类型的现有方法却存在明显的缺点: 如果在上述第四步, 一个信道当与其父信道合并而被删除或由于停止或跳转事件而被取消, 则那些正在使用这个被删除的信道的子信道的客户不得不改变它们正在使用的信道。例如, CT 方案简单地选择仍在系统中的较早视频流中的最近时间的视频流信道作

为下一个合并目标，而即使没有新的子信道被建立，CT 计算的该合并目标也不一定是可到达的，原因是该目标流信道本身在被其后续信道追上以前可能已经与该目标信道的目标信道进行了合并。当这种情况发生时，后续流信道必须再次使用 CT 算法选择一个新的合并目标。另外，目标流信道的任意

5 停止、暂停、快进/后退等操作都可能使后续流信道的合并不能发生，并迫使它们重新选择新的父信道。

为了将合并树的改变通知给受到影响的客户，视频服务器必须主动地向这些客户中的每一个发送通知。这可能导致以下几种负面影象：

- 10 1. 由于通知的数量与受到影响的客户端的数量和未预料到的信道停止事件的频率成正比，从视频服务器到客户端的反向通知大大地增加了视频服务器的负载。
2. 客户必须准备接收来自于互联网的未知区域的进入连接，而这增加了客户端受其它意外影响的可能性。
- 15 3. 该反向通知可能不能通过具有一定配置的防火墙。例如，如果防火墙内的客户试图观看存储于该防火墙外的视频服务器中的一段视频片段，则该服务器永远都不会向客户端传送通知。

### 发明内容

为了解决上述问题，通过使用响应于开始、跳转、合并和停止事件而决定合并路径的准则，本发明提供一种视频点播系统中的信道合并方法，所述方法包括步骤：（1）响应于多个客户对某一视频节目的请求，建立一根信道（S1）和至少一个子信道（S11），所述根信道（S1）根据最早发出请求的客户机的请求而建立，所述多个子信道（S11）中的每个对应于一较晚发出请求的客户机的请求而建立；以及（2）对于所建立的每一个信道，监测使用

20 该信道的客户机的数目变化，如果使用被监测信道的客户机数目不为零，则维持该信道，如果使用被监测信道的客户机数目为零，则关闭该信道。

本发明还提供一种用于视频点播系统的信道合并装置，所述信道合并装置配置于所述视频点播系统中的视频服务器中或与其可操作地连接，其中所述信道合并装置包括：信道选择单元，用于响应多个客户端对某一视频节目的

30 请求，建立一根信道（S1）和至少一子信道（S11），所述根信道（S1）根据最早发出请求的客户端的请求而建立，所述多个子信道（S11）中的每一

个对应于一较晚发出请求的客户端的请求而建立；信道控制单元，用于对所建立的每一个信道，监测使用该信道的客户端的数目变化，如果使用被监测信道的客户端数目不为零，则维持该信道，如果使用被监测信道的客户端数目为零，则关闭该信道。

- 5        在本发明中，所有的信道“合并事件”都是从最次级子信道向根信道的方向进行的，因此不会发生客户端正在使用的信道被移除的情况。即使直接发生停止事件，在使用某个信道的所有客户端（以该信道的子信道的方式）明确地释放该信道之前，该信道也不会被移除。所以避免了反向通知，从而一个客户端的行为不会影响其它的客户端，并减小了视频服务器和网络的负载。
- 10       同时，因为当客户端触发的事件发生时动态地构建和调整合并树，而每个客户端的合并路径是基于合并树而进行动态调整的，所以这种方法固有地支持诸如播放、停止、搜索（快进/后退）等的 VCR 功能。

- 15       本发明的另一个主要优点是控制视频服务器的方法与 HTTP（超文本传输协议）的请求/响应模式兼容，因此可以容易地在 HTTP 上实现。HTTP 是在互联网上交换应用数据的主流协议，并且网络服务的出现进一步强化了基于互联网的应用在数据传输方面应该尽可能依赖 HTTP 的趋势。另外，大多数防火墙在允许一般数据流量的通过方面是难于处理的，而不是 HTTP。

#### 附图说明

- 20       通过结合附图对本发明进行的详细描述，将使本发明的上述优点和其它特点变得更加清楚。

图 1 是示出具有视频服务器和多个客户端的组播网络。

图 2 是示出进行信道合并的视频节目流的示意图。

图 3 是示出信道合并树的示意图。

- 25       图 4 是在图 1 的视频服务器和客户端之间发生的请求/应答的时序图。

图 5 是根据本发明配置于视频服务器中的信道合并装置的结构图。

图 6 是根据本发明发生“开始事件”时视频服务器的处理的流程图。

图 7 是根据本发明发生“跳转事件”时视频服务器的处理的流程图。

图 8 是根据本发明发生“合并事件”时视频服务器的处理的流程图。

- 30       图 9 是根据本发明发生“停止事件”时视频服务器的处理的流程图。



具体实施方式

下面将结合附图对本发明进行详细地描述。

除非特别说明，在本说明书的下文中涉及到的开始、停止、跳转、和合并等操作均是对于播放相同视频节目（对象）的各个信道进行的。

5 图 1 是具有服务器 1 和多个客户端 4 的组播网络。该组播网络包括视频服务器 1、互联网 2、防火墙 3 和多个客户端 4，其中视频服务器 1 与客户端 4 通过互联网 2 和防火墙 3 进行通信。在本发明中，客户端 4 通过防火墙 3 和互联网 2 向视频服务器 1 发出请求，请求对某段视频节目进行播放、停止、暂停、快进/后退等诸如 VCR 功能的操作。

10 下面参考图 4 说明图 1 中的视频服务器 1 和客户端 4 之间所发生的请求/应答操作。

图 4 是在视频服务器 1 和客户端 4 之间发生的请求/应答的时序图。

在图 4 中，每个消息都包含要执行的动作的消息类型，该消息类型由一系列参数来指定。这里定义了五种消息类型：打开（OPEN）、播放（PLAY）、  
15 暂停（PAUSE）、合并（MERGE）、和关闭（CLOSE）。对于由客户端发送的每个消息，服务器都会发送回应答（RESPONSE）消息。

在步骤（1）中，通过发送 OPEN 消息给视频服务器 1，客户端 4 与视频服务器 1 建立一个会话，该 OPEN 消息包含唯一识别视频服务器 1 上所请求的视频文件（视频节目）的视频标识（Video ID）。如果成功地建立了该会  
20 话，则视频服务器 1 将会发送回包含信道信息的 RESPONSE 给客户端 4，该信道信息例如是组播地址和端口号，客户机 4 可以根据该信道信息接收所请求的视频节目。

在步骤（2）中，客户端 4 发送一个 PLAY 消息，以请求开始播放该视频节目，或从暂停状态进行恢复，在该 PLAY 消息中可以指定一个偏移参数，  
25 以搜索该视频节目的指定位置。在该 RESPONSE 消息中可以包含给客户端 4 的附加信息，如指示客户机 4 接收所请求的节目等的信息。

在步骤（3）中，当客户端 4 检测到发生了信道合并时，发送 MERGE 消息给视频服务器 1（当然，视频服务器 1 也可以通过自己的诸如信道控制单元 20 的部件对信道进行计算而得出信道合并事件的发生）。该视频服务器  
30 1 将关闭不再使用的信道，并发送回指示客户端 4 应加入的附加信道。

在步骤（4）中，客户端 4 可以发送 PAUSE 消息给视频服务器 1，以在

播放视频节目时暂时停止数据传输，并且视频服务器 1 进行相应的响应。

在步骤 (5) 中，客户端 4 可以发送 CLOSE 消息给视频服务器 1，以关闭与视频服务器 1 的会话，并且视频服务器 1 进行相应的响应。

如果我们将上述所有的请求和通知消息都模型化为各个事件，则在视频服务器 1 中共有四种事件：即“开始事件”、“跳转事件”、“停止事件”、和“合并事件”。这些事件都可以通过上述五种消息类型，即打开 (OPEN)、播放 (PLAY)、暂停 (PAUSE)、合并 (MERGE)、和关闭 (CLOSE) 进行操作。

当客户端 4 发送关于对象 (视频节目) 的请求时发生“开始事件”，即客户端 4 在时刻  $t$  请求播放一段视频节目 (用 PLAY 消息)；当客户端 4 发送关于对象的快进或后退请求时发生“跳转事件”，当发生“跳转事件”时，相当于客户端 4 向视频服务器 1 发送了从时刻  $t+s$  或  $t-s$  播放该视频节目的请求，其中  $s$  是跳转的对象 (即所请求的视频节目其它部分) 相对于在时刻  $t$  的偏移时间)，这时，视频服务器 1 建立一个新的信道以从时刻  $t-s$  播放该视频节目 (用 PLAY 消息)，同时关闭从时刻  $t$  开始播放该视频节目的信道 (用 CLOSE 消息，对此操作将在下文中进行详细的说明)；当客户端 4 不再需要一个对象时发生“停止事件”，即关闭发送该视频节目的信道 (用 CLOSE 消息)；当客户端 4 到达该客户端 4 正在收看的最后信道对 (channel pair) 已经成功合并的合并点时，发生“合并事件” (用 MERGE 消息和 CLOSE 消息)，如在本发明背景技术中所述的合并方法中所述的情况，即图 2 中的起始点 (b) 为该合并点。

为了实现上述对某段视频节目进行播放、停止、暂停、快进/后退等诸如 VCR 功能的操作，本发明提供了信道合并装置 40。

下面结合图 5 说明根据本发明的信道合并装置 40 的具体结构。

图 5 是根据本发明的信道合并装置 40 的结构图。

本发明的信道合并装置 40 配置于视频服务器 1 的内部，包括：信道选择单元 10，用于接收多个客户端对某一视频节目的请求，并响应该请求建立一根信道 (S1) 和至少一子信道 (S11)，所述根信道 (S1) 根据最早发出请求的客户端的请求而建立，所述多个子信道 (S11) 中的每一个对应于一较晚发出请求的客户端的请求而建立，并且该信道选择单元 10 还在信道合并过程中为一子信道寻找满足后面将描述的条件表达式 (1) 和 (2) 的父信

道；和信道控制单元 20，用于根据客户端 4 的请求和信道选择单元 10 的选择结果等执行信道的建立、合并和关闭等操作。

另外，上述信道合并装置 40 可以与视频服务器 1 可操作地连接在一起，而不必要配置在视频服务器 1 的内部。同时，信道选择单元 10 和信道控制单元 20 还可以是同一个单元，例如可以是计算机中的 CPU（中央控制单元），用于执行存储于该计算机中的 ROM 或 RAM 或其它存储介质（未示出）中的可执行程序，以实现与信道选择单元 10 和信道控制单元 20 相应的功能。

该信道控制单元 20 还包括一计数单元 22，该计数单元 22 用一计数参数（ref\_num）来标记使用每个信道的客户端 4 的数量，以实现信道控制单元 20 的控制功能。当所述每个信道及其子信道发生合并、跳转、或停止事件时，所述计数单元 22 减小计数参数的值，并且如果计数参数的值等于零，则在视频服务器 1 侧关闭其计数值等于零的信道。

如果该计数参数的值不等于零，则在服务器端 1 保持该信道，而执行了合并、跳转、或停止事件的所述客户端则不再接收该信道播放的节目。

在本发明中，视频服务器 1 中的信道选择单元 10 响应于多个客户机 4 对某一视频节目的请求，如图 3 所示，建立一根信道 S1 和至少一个子信道（S11），该根信道（S1）是根据最早发出请求的客户机（例如 A）的请求而建立的，而该多个子信道 S11（如还有 S5 和 S6 等）中的每个是对应于一较晚发出请求的客户机的请求而建立的，该根信道 S1 与多个子信道 S11（和 S5 和 S6 等）形成树结构。当然，上述请求都是满足视频点播条件的请求。

所有上述信道都传送基于客户端 4 的请求而来自视频服务器 1 的组播流，并且每个信道的组播流都可以由所有的客户端 4 接收到。每个客户端 4 最多可同时接收两个信道，其中一个信道是为该客户端 4 本身而发起的，而另一个信道则是为之前的客户端而发起的较早的信道，例如，请求对象（视频节目）的第一个客户端只从信道 s1（如图 3 所示）接收视频流，而第二个客户端同时从信道 s1 和 s2 接收该视频流，其中 s1 被选择作为 s2 的父信道（如图 3 所示）。每个客户端都必须能够在本地存储器（未示出）上存储所接收的视频流。对于所建立的每一个信道，视频服务器 1 通过本发明的信道控制单元 20 监测使用该信道的客户机 4 的数目变化，如果使用被监测信道的客户机 4 数目不为零，则维持该信道，如果使用被监测信道的客户机 4 数目为零，即已经没有客户机 4 在使用该信道了，则视频服务器 1 通过本发

明的信道控制单元 20 关闭该信道。

这里，视频服务器 1 中的信道控制单元 20 是通过自己的计算或通过接收来自发生信道合并的客户端 4 的消息来检测两个信道（视频流）的合并的。并且，该合并过程一直持续到所有的子信道都合并进根信道（对于同一视频  
5 节目所建立的第一个信道）为止。这里假设视频节目的长度是无限的。

下面参考附图详细说明根据本发明的信道合并的方法。

还参考图 3，图 3 是示出信道合并树的示意图。其中的每个节点都表示一个信道，在这里将对某一视频节目第一个建立的信道定义为根信道，将较高一级的信道定义为较低一级的信道的父信道，相反，将较低一级的信道定义为较高一级的信道的子信道。如图 3 所示，S1 是根信道，S6 是 S11 的父  
10 信道，而 S11 是 S6 的子信道，同时，S6、S10、S11、和 S12 是 S5 的子孙信道的集合。

如果视频服务器 1 中的信道控制单元 20 在客户端 4 的请求（信道的开始、跳转）下建立了一个新的信道，我们假设它是信道 S11，则信道选择单元 10 立即为该信道 S11 寻找其父信道 S6。如果找到了父信道 S6，则返回该  
15 父信道 S6；否则，返回“找不到父信道”的消息。

上述操作执行的方法如下：

步骤 1：通过下列表达式（1），从信道 S11 可能并入的活动根信道的集合中找到一个最近的根信道，如 S1，即：

$$20 \quad \min(S11.start\_time - S1.start\_time) < object\_length/2 \quad (1)$$

其中， $\min(S11.start\_time - S1.start\_time)$  表示信道 S11 的开始时间（ $S11.start\_time$ ）与根信道集合中的各个根信道的开始时间（ $S1.start\_time$ ）的差值中的最小值， $object\_length/2$  表示所播放的视频节目的总长度（总时间）的 1/2。其中，开始时间（ $start\_time$ ）表示该  
25 信道开始的时刻。因此，上述表达式的含义是：信道 S11 的开始时刻与根信道集合中的某个根信道 S1 的开始时刻的差值中的最小值要小于所播放的视频节目的总长度（总时间）的 1/2。在这种情况下，我们认为该根信道 S1 是可以被其子信道 S11 追上的，可以作为信道 S11 的根信道 S1；否则，如果上述开始时刻的差值大于所播放的视频节目的总长度（总时间）的 1/2，  
30 我们就认为这个根信道是不可以达到的，从而返回“不能找到根信道”的消息，并将该信道 S11 作为一个新的根信道。

如果用上述条件找到了有效的根信道 S1，则前进到下面的步骤 2。

步骤 2: 如果将根信道 S1 的子孙信道的集合定义为 S，这里所述的子孙信道包括其直接父信道就是根信道 S1 的子信道 S5 和信道 S5 的子信道 S6，这里定义信道 S11 是信道 S6 的子信道，同时可知信道 S11 也是信道 S5 的孙信道，依次类推。通过下述条件表达式 (2) 在集合 S 中找出信道 S6 (其中信道 S6 是信道 S11 的父信道，而信道 S6 和信道 S11 均在子孙信道集合 S 中，且为根信道 S1 的子孙信道):

$$\min(S11.start-time-S6.start-time) < S6.start-time-S5.start-time \quad (2)$$

上述表达式的含义为: 为信道 S11 找一个其将要并入的父信道 S6，而这个父信道 S6 要满足一个条件，该条件就是: 该信道 S11 的开始时刻与待选父信道 S6 的开始时刻的差值中的最小值应该小于该父信道 S6 的开始时刻与该父信道 S6 的父信道 S5 的开始时刻的差值。

如果满足上述条件的信道 S6 存在，则将信道 S6 返回作为信道 S11 在下一轮合并中的父信道。否则，如果找不到满足上述条件的任何信道，则将根信道 S1 返回作为信道 S11 的父信道。

很明显，只要满足上述条件，也就保证了在信道 S11 并入其父信道 S6 之前，该父信道 S6 不会并入该父信道 S6 的父信道 S5。

也就是说，在一个信道 S6 合并进其父信道 S5 的时刻，该信道 S6 的各个子信道，如 S10 和 S11，均已经合并进其父信道 S6。因此，从通过计算或通过接收来自客户端 4 的消息而得知该信道 S6 合并进其父信道 S5 的时刻起，视频服务器 1 关闭该信道 S6，从而不会影响其它客户端的使用，因为该信道 S6 的选择如果满足上述条件，在这个时刻已经没有其它客户端在使用该信道 S6 了，即满足上述条件的信道 S10 和 S11 均已经合并进入信道 S6 了。这样就保证了所有的信道合并都从最次级子信道向根信道的方向进行的，即在使用某个信道的所有客户端 4 (以该信道的子信道的方式) 明确地释放该信道之前不移除该信道。

下面结合图 6、图 7、图 8、和图 9 说明响应于开始、跳转、合并、和停止这四种类型的事件，视频服务器 1 及其中的各单元的工作情况。

首先说明响应于开始事件，视频服务器 1 及其中的各单元的工作情况。

如图 6 所示，当客户端 4 在时刻 t 调用“开始事件”(即客户端选择播放某一段视频节目)时:

在步骤 100, 信道选择单元 10 创建一个新的信道 S11 播放该视频节目, 并在该信道 S11 中设置  $\text{start\_time}=t$ ,  $\text{object\_offset}=0$  (对象偏移时间: 表示该信道开始时视频节目的偏移时间, 即该信道的开始时间相对于在时刻  $t$  开始的信道的偏移时间),  $\text{ref\_num}=1$  (参考数值: 表示客户端的数量); 这里  $\text{start\_time}=t$  表示该信道是从时刻  $t$  开始的,  $\text{object\_offset}=0$  表示该信道没有偏移,  $\text{ref\_num}=1$  表示只有一个客户端正在使用该信道。

在步骤 102, 通过信道选择单元 10 查找该信道 S11 的父信道。

在步骤 103, 判断是否存在该父信道。

在步骤 104, 如果没有找到其父信道, 则信道合并单元 20 将这个信道 S11 作为新的根信道 (即设置  $\text{root\_flag}=1$  (根标志: 表示该信道是否是根信道的标志)), 并且这个信道 S11 是客户端 4 应该收看的唯一信道。

否则, 在步骤 105, 如果找到了父信道 S6, 则该客户端 4 必须同时收看该信道 S11 和它的父信道 S6。

在步骤 106, 向客户端 4 发送上述操作信息, 以响应客户端 4 触发的开始请求。

下面说明响应于跳转事件, 视频服务器 1 及其中的各单元的工作情况。

如图 7 所示, 当客户端 4 在相对于时刻  $t$  的对象偏移时间  $s$  调用“跳转事件” (即客户端 4 在  $t-s$  的时刻进行了相当于 VCR 功能的快进或后退的操作) 时:

在步骤 200, 信道选择单元 10 创建一个新的信道 S11 以播放在时刻  $t-s$  开始的该视频节目, 并在该信道 S11 中设置  $\text{start\_time}=t-s$ ,  $\text{object\_offset}=s$ ,  $\text{ref\_num}=1$ ; 这里  $\text{start\_time}=t-s$  表示该信道是从时刻  $t-s$  开始的,  $\text{object\_offset}=s$  表示该信道的开始时间相对于在时刻  $t$  开始的信道的偏移时间为  $s$ ,  $\text{ref\_num}=1$  表示只有一个客户端正在使用该信道。

在步骤 202, 假设在时刻  $t$  开始的信道为 S4, 则减少信道 S4 的计数参数的值, 即由于客户端 4 刚才使用的信道 S4 已经被执行了停止操作, 因此目前正在使用信道 S4 的客户端的数量减少了一个。这时, 如果信道 S4 的计数参数为零, 表示目前已经没有客户端在使用该信道 S4 了, 则信道控制单元 20 关闭该信道 S4; 相反, 如果这时信道 S4 的计数参数不为零, 表示目前还有客户端 4 在使用该信道 S4, 则不能关闭该信道 S4, 以供正在使用该信道 S4 的其它客户端 (如 S8 和 S9) 继续使用, 但是调用“跳转事件”的

该客户端 4 则不再使用该信道 S4，而转向信道 S11 收看该视频节目了。

在步骤 203，通过信道选择单元 10 查找该新的信道 S11 的父信道。

在步骤 204，判断是否存在该父信道。

在步骤 205，如果没有找到父信道，那么这个信道 S11 就作为新的根信道（即设置 `root_flag=1`），并且这个信道 S11 是客户端 4 应该收看的唯一信道。

否则，在步骤 206，如果找到了父信道 S6，则该客户端 4 必须同时收看该信道 S11 和它的父信道 S6；

在步骤 208，向客户端 4 发送上述操作信息，以响应客户端 4 触发的跳转请求。

下面说明响应于合并事件，视频服务器 1 及其中的各单元的工作情况。

如图 8 所示，当客户端 4 触发了“合并事件”（即发生子信道 S11 到父信道 S6 的合并）时：

在步骤 300，信道控制单元 20 减少子信道 S11 的计数参数的值，即由于客户端 4 正在使用的信道 S11 已经合并入其父信道 S6，因此目前正在使用信道 S11 的客户端的数量减少了一个。这时，如果信道 S11 的计数参数为零，表示目前已经没有客户端在使用该信道 S11 了，则视频服务器 1 关闭该信道 S11；相反，如果这时信道 S11 的计数参数不为零，表示目前还有客户端在使用该信道 S11，则不能关闭该信道 S11，以供正在使用该信道 S11 的其它客户端继续使用，但是调用“合并事件”的该客户端 4 则不再使用该信道 S11，而是转向了信道 S6 收看该视频节目了。信道 S11 的计数参数的值就是正在使用该信道 S11 的客户端 4 的数目，例如，如果计数参数是 1（即 `ref_num=1`），表示有一个客户端 4 正在使用该信道 S11，如果计数参数是 5（即 `ref_num=5`），则表示还有五个客户端 4 正在使用该信道，而这些信道均是信道 S11 的次级子信道。

在步骤 302，通过信道选择单元 10 查找该信道 S6 的父信道。

在步骤 303，判断是否存在该父信道。

在步骤 304，如果没有找到该信道 S6 的父信道，那么这个信道 S6 就作为新的根信道（即设置 `root_flag=1`），并且这个信道 S6 是客户端 4 应该收看的唯一信道。

否则，在步骤 305，如果找到了信道 S6 的父信道 S5，则该客户端 4 必

须同时收看该信道 S6 和其新认定的父信道 S5。

在步骤 306, 向客户端 4 发送上述操作信息, 以响应客户端 4 触发的合并请求。

下面说明响应于停止事件, 视频服务器 1 及其中的各单元的工作情况。

5 如图 9 所示, 当客户端 4 调用“停止事件”(即客户端执行了相当于 VCR 功能的停止操作)或对象(视频节目)到达终点(即结束)时:

假设客户端正在使用信道 S11 观看一视频节目, 则在步骤 400, 判断该视频节目是否已经到达终点, 即是否已经结束。

10 如果对象(视频节目)已经到达终点, 则信道控制单元 20 在步骤 402 关闭客户端 4 正在使用的信道 S11, 并直接释放该信道的所有资源。

15 如果该视频节目未到达终点, 则在步骤 404, 则信道控制单元 20 如上所述减少信道 S11 的参考数值的值。如果信道 S11 的参考数值为零, 则关闭该信道 S11 并释放该信道的所有资源; 相反, 如果信道 S11 的参考数值不为零, 则信道控制单元 20 不关闭该信道 S11, 以供正在使用该信道 S11 的其它客户端继续使用, 而调用“停止事件”的客户端则不再使用该信道 S11。

20 本发明上述控制视频服务器的方法与 HTTP(超文本传输协议)的请求/响应模式兼容, 因此可以容易地在 HTTP 上实现。HTTP 是在互联网上交换应用数据的主流协议, 并且网络服务的出现进一步强化了基于互联网的应用在数据传输方面应该尽可能依赖 HTTP 的趋势。另外, 大多数防火墙在允许一般数据流量的通过方面是难于处理的, 但是对于 HTTP 来说, 则不存在这个问题。

上面对本发明的实施例进行了详细地说明。本领域的普通技术人员应该明白, 按照本发明的精神及指导思想对本发明做出的各种修改都在本发明后附的权利要求书所要求保护的范围内。



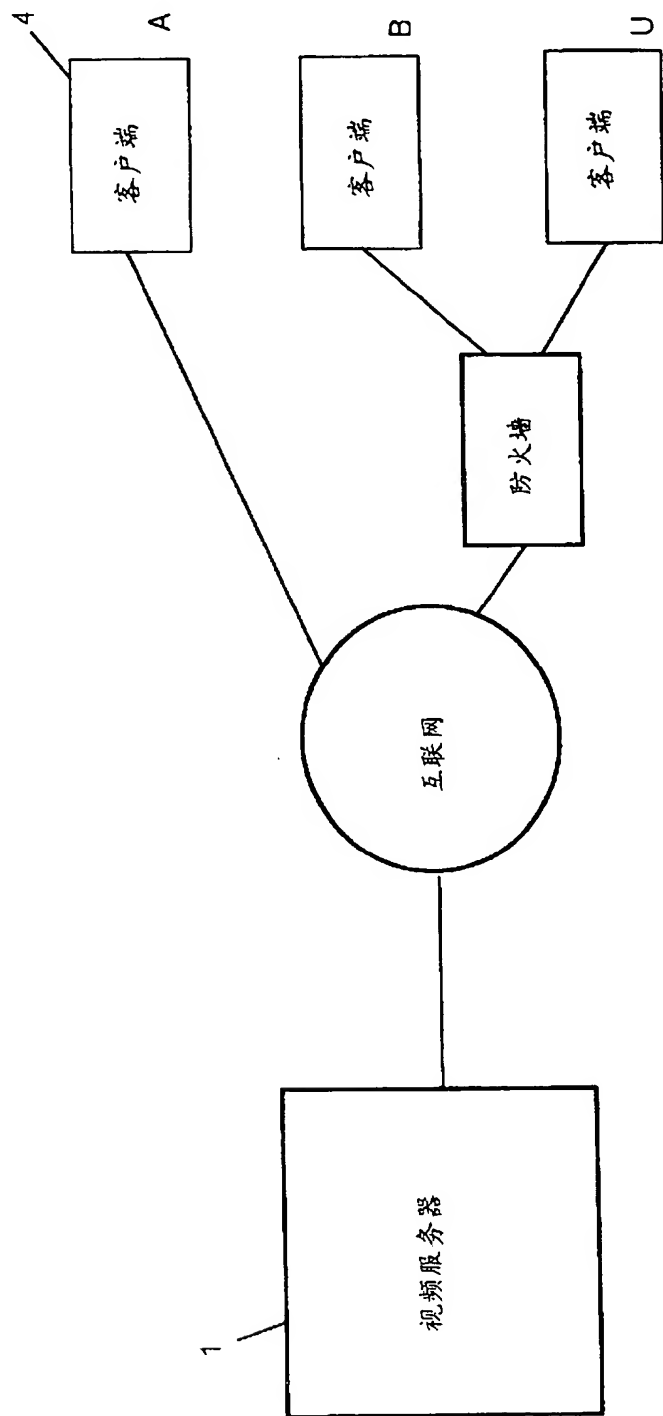


图 1

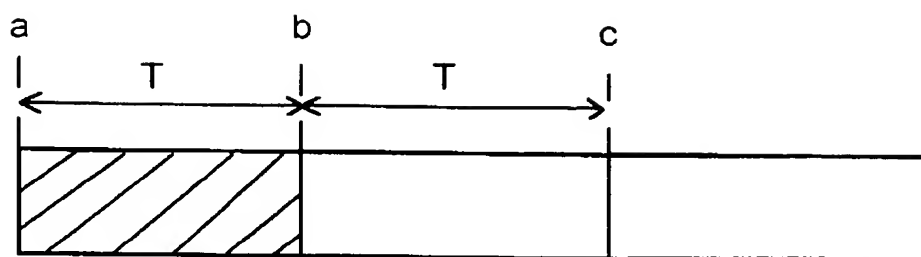


图 2

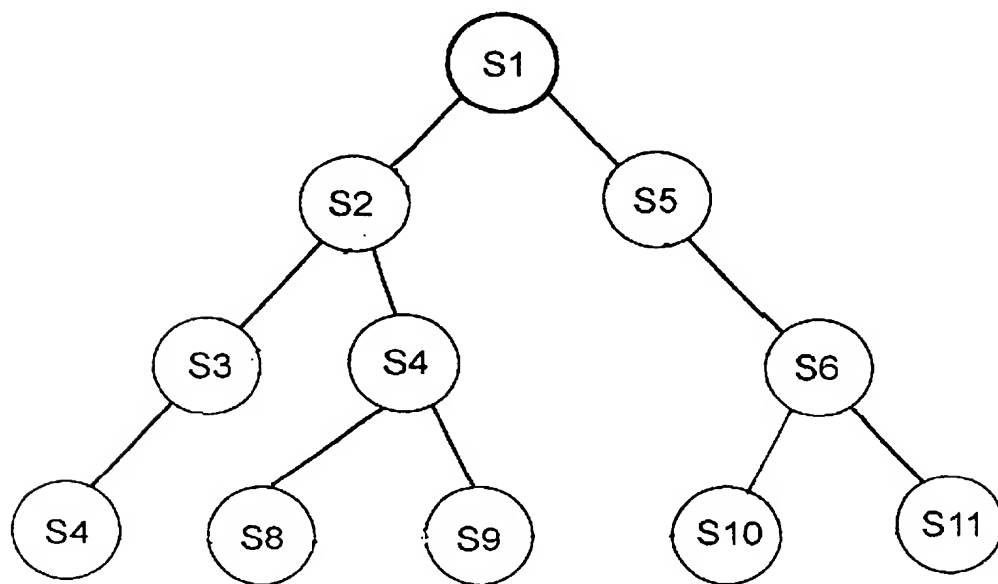


图 3

73

客户端

服务器

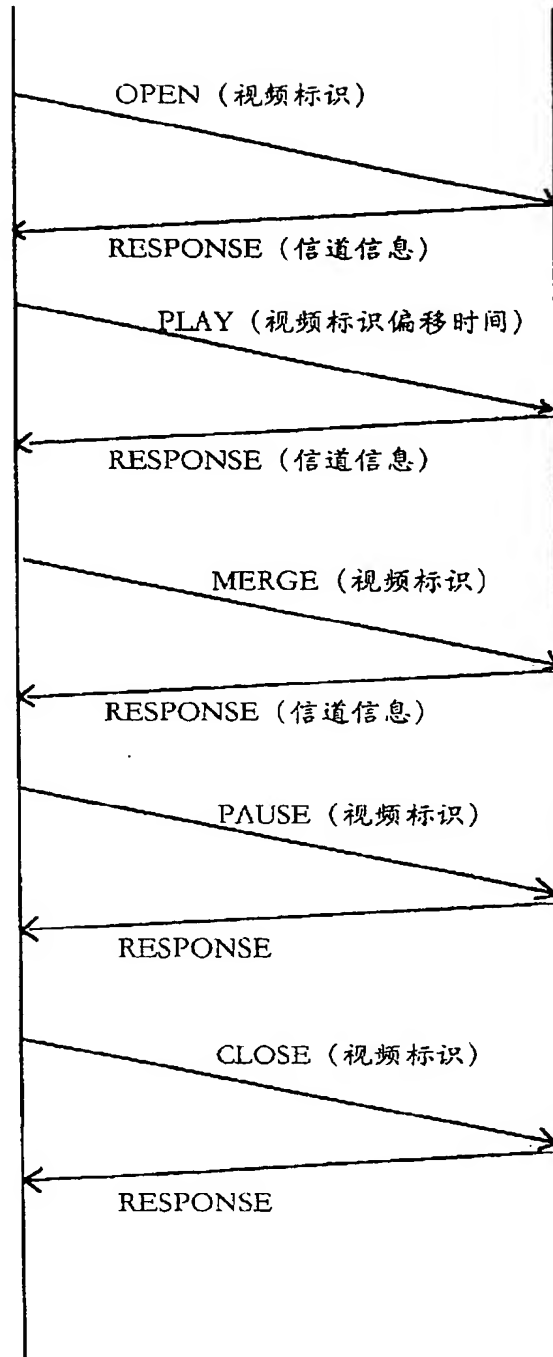


图 4

54

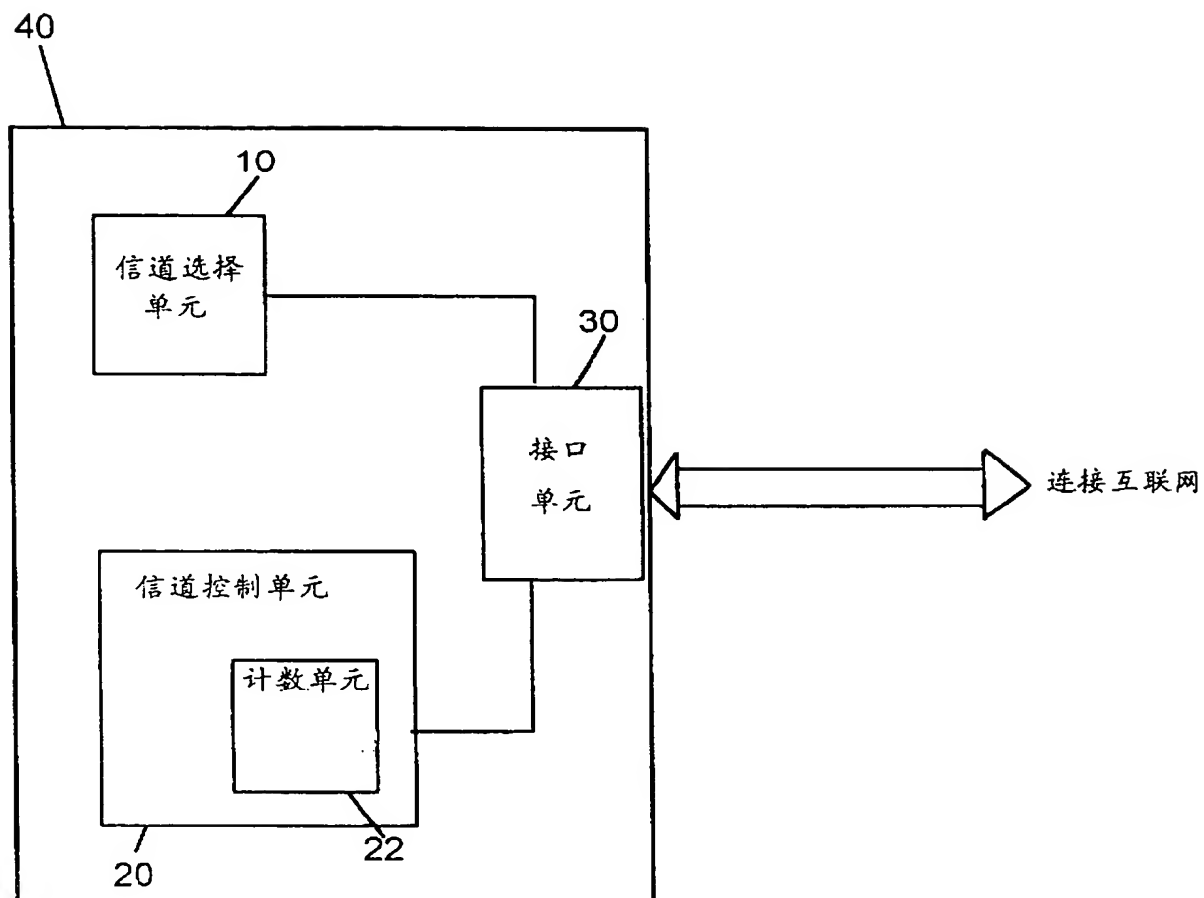


图 5

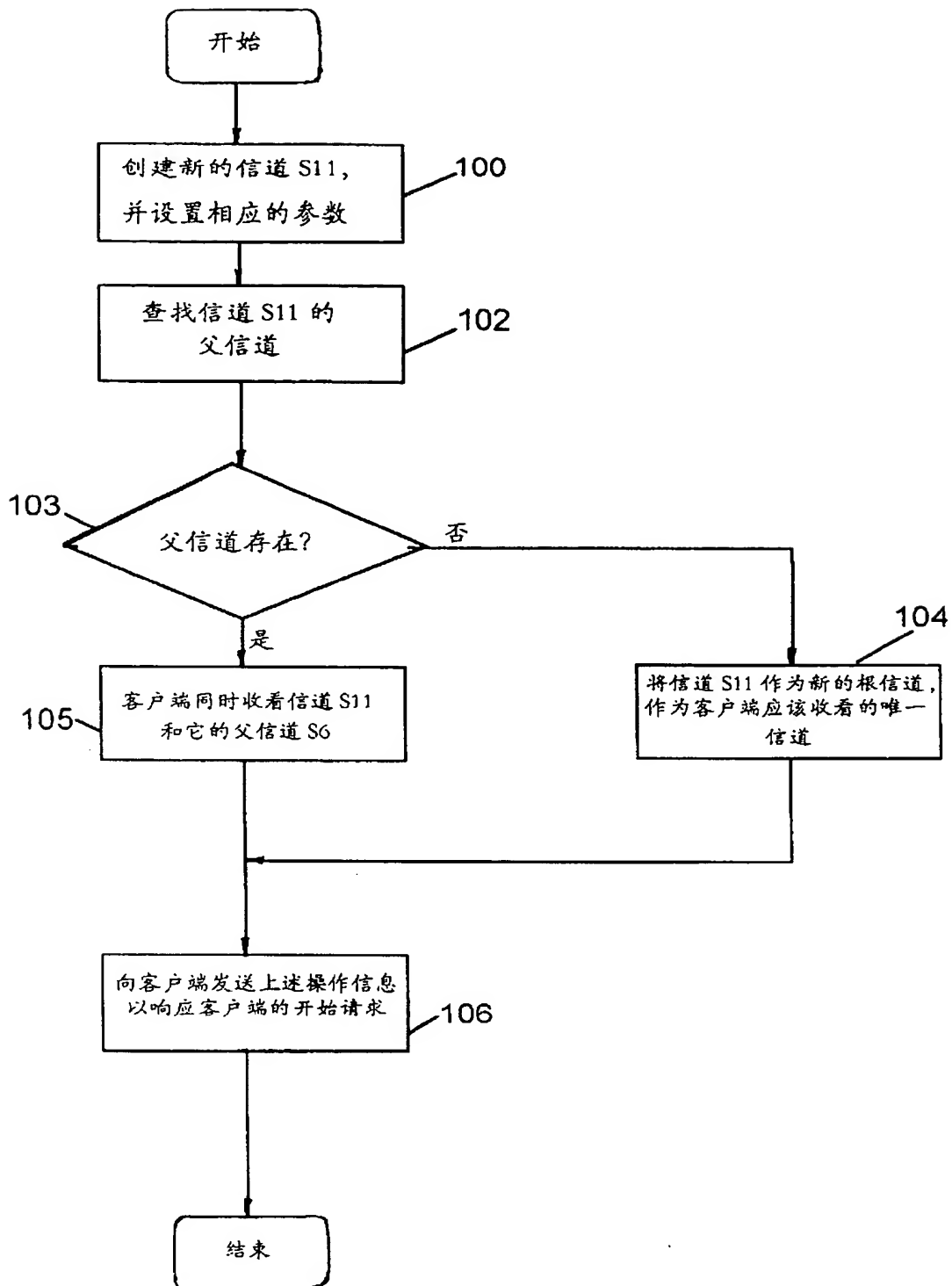


图 6

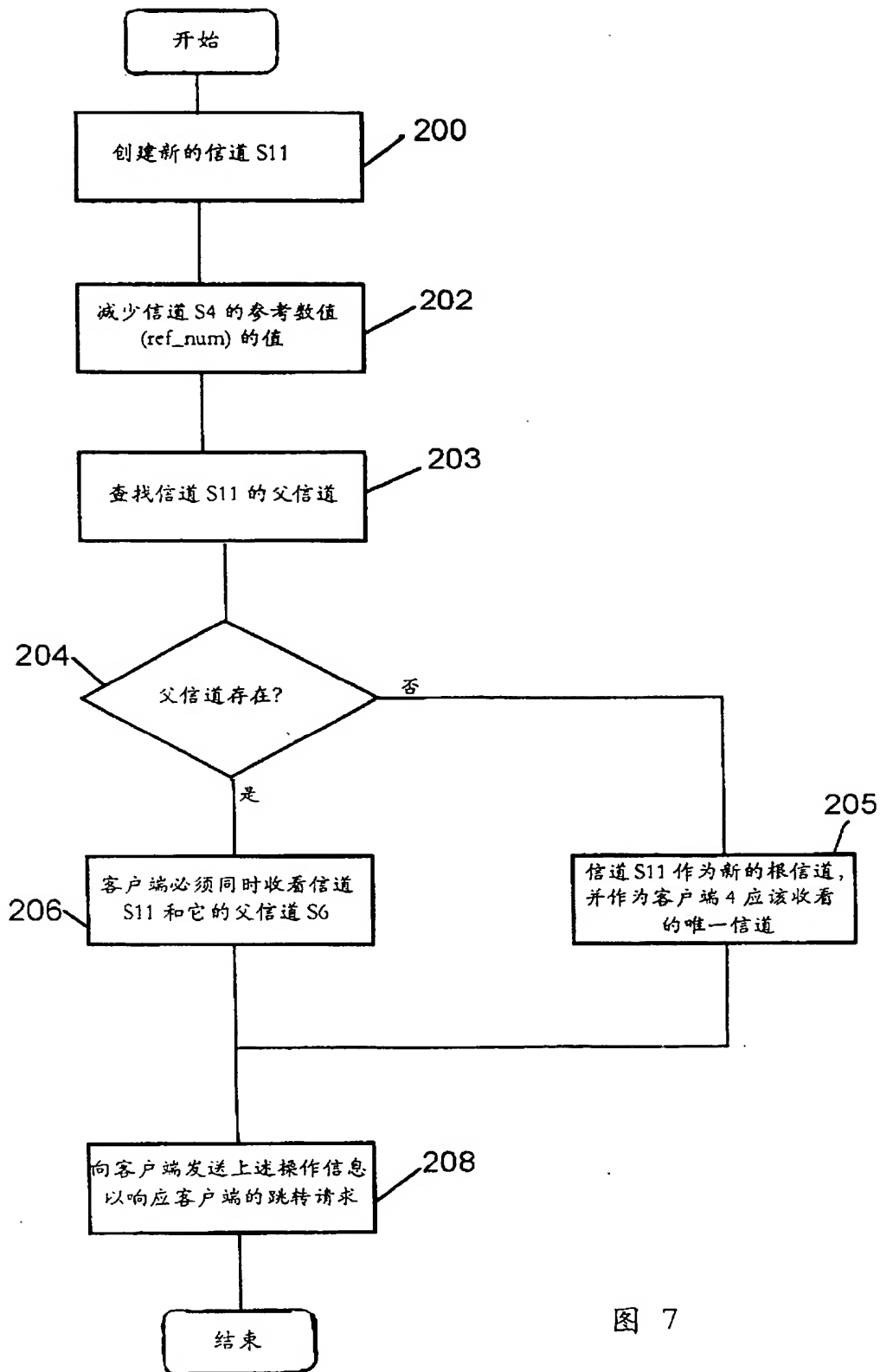


图 7

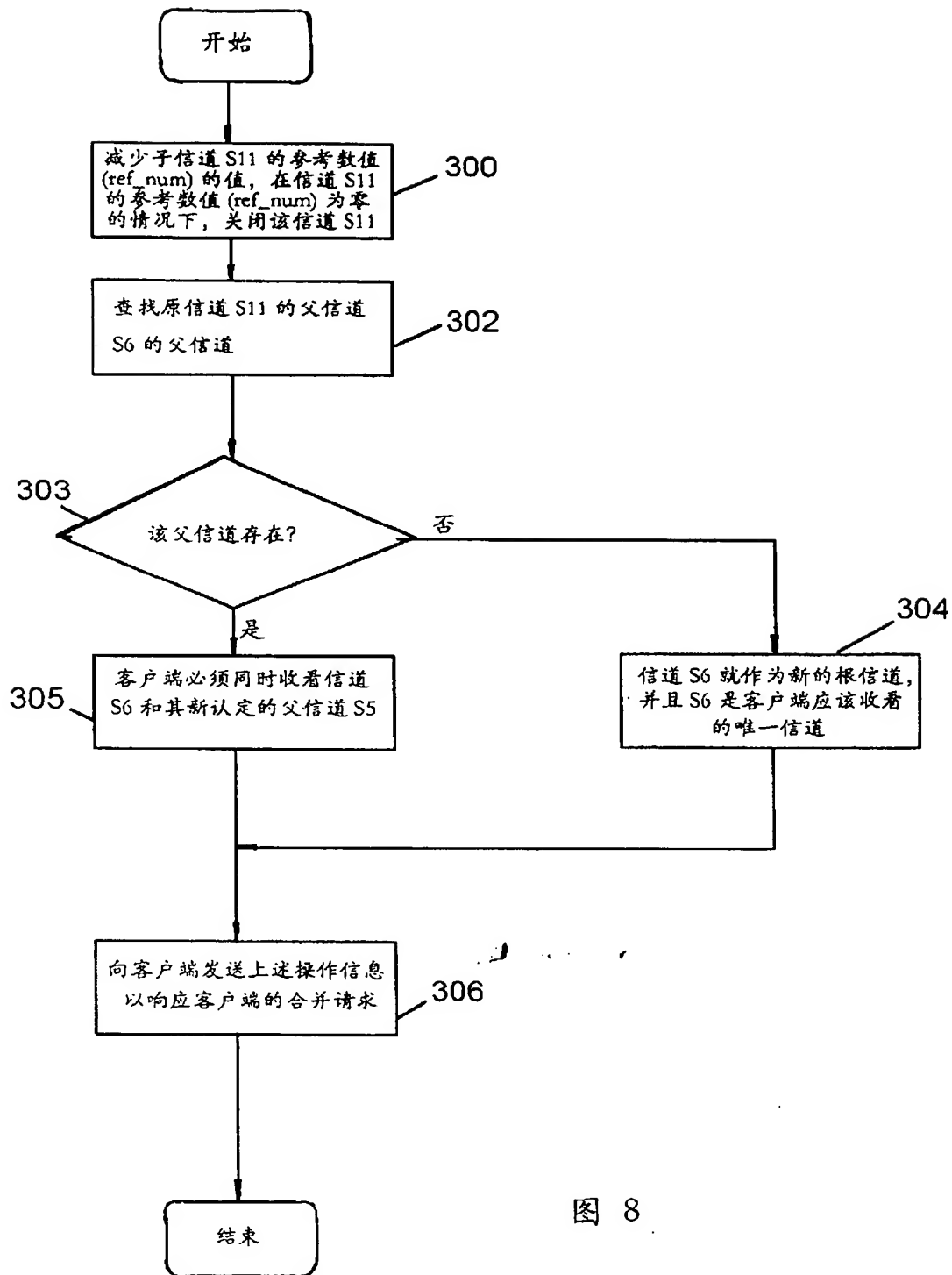


图 8

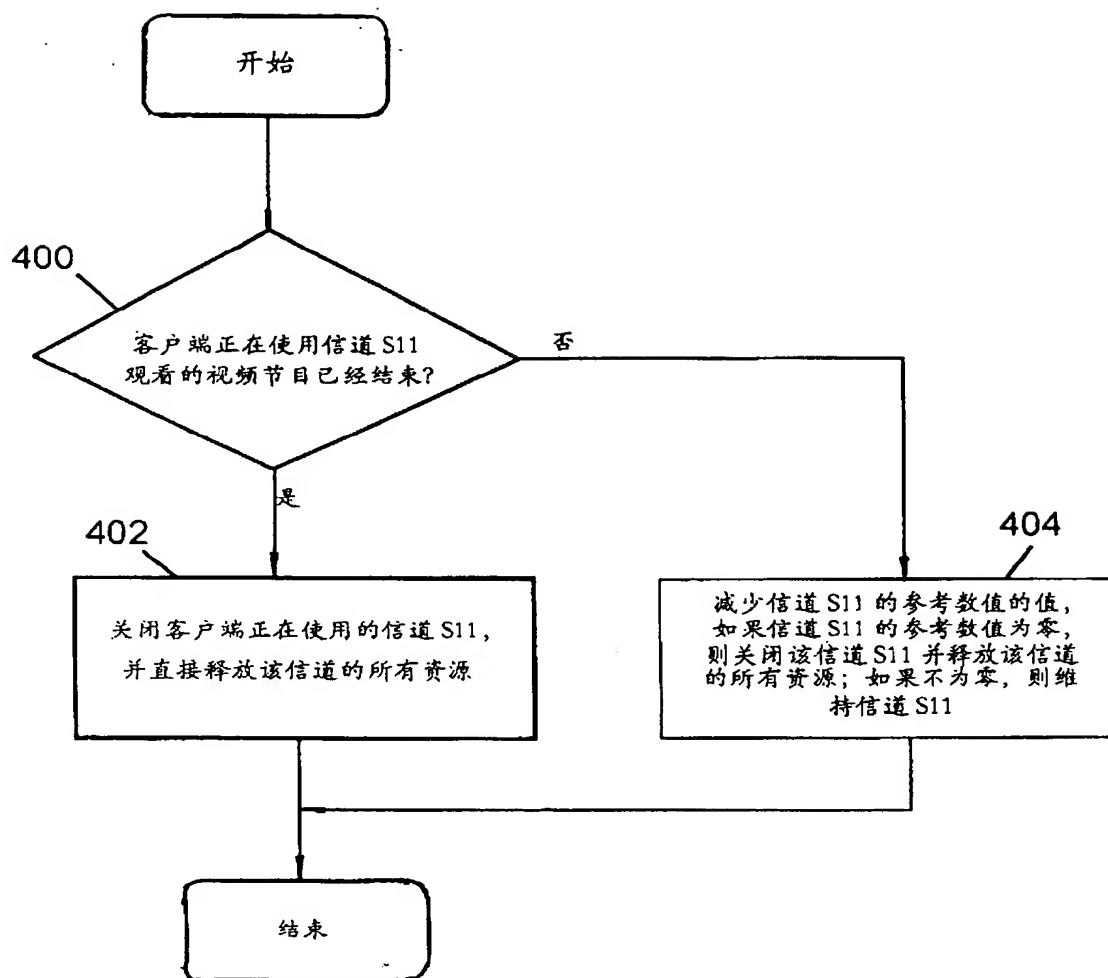


图 9